Visual Grid Programming for Scientists

Keith R. Jackson, LBNL

Gregor von Laszewski, ANL

Despite the ongoing investment in collaboration and Grid technologies, scientists are frequently unable to easily harness the power of Grid and collaborative technologies without considerable effort. The level of effort currently required is one of the major barriers to widespread adoption of Grid technologies amongst the scientific community. To help ensure the relevance of the National Collaboratories program it is essential to research and develop methods of interacting with Grid middleware that easily support the scientific process. The use of visual programming techniques to provide an easy to use, dynamic, domain specific interface to Grid middleware will be an important part of the tools necessary to enable the widespread usage of Grids.

The computing demands of many state-of-the-art scientific applications—such as climate modeling, astrophysics, high energy physics, structural biology, chemistry, and tele-immersive engineering—requires the coordinated use of numerous distributed and heterogeneous components, including advanced networks, computers, storage devices, display devices, and scientific instruments. Such a national collaborative Grid infrastructure is being developed and supported by DOE, NSF, and NASA.

Developing advanced scientific *applications* for these emerging national-scale computational Grids is still a difficult task, however. Although elementary Grid services exist that enable scientific application developers to authenticate, access, manage, and discover sophisticated remote resources, these frameworks are not compatible with the commodity technologies and frameworks used by application scientists today. Additionally the technologies often are too complex to be used by computational scientists, who may lack the expertise needed for dealing with such a complex infrastructure. A higher level of abstraction is demanded that supports easy access to the Grid using tools that the application scientists are comfortable with.

Currently many scientists are familiar with visual programming environments. Visual programming is a form of software programming environment in which program logic, input and outputs, and execution are described graphically, rather than using lines of code. Visual programming environments have been applied to a number of contexts, with data acquisition and visualization being the most established. The LabView software, from National Instruments, and the AVS software from Advanced Visual Systems are two common commercial visual programming environments that are widely used by the scientific community.

The visual programming (VP) model provides a natural interface for

specifying ordered flows of data through computational processing modules. Visual programming environments contains all the generic idioms for composing groups of tasks and tracking of the progress of a complex task, and thus are capable of acting as a workflow description and execution framework. In particular, the visual nature of VP lends itself to data analysis and visualization. Scientists can apply reduction, filtering, and mapping to raw data using standard data processing modules, then apply a data visualization module.

Figure 1 shows a workflow that represents a common collection of tasks performed during a molecular dynamics simulation of a protein. Each box is a "visual node" representing a particular domain-specific application with well-defined inputs. A yellow triangle connecting a labeled envelope to a node represents static parameter input. Purple rectangles on the left hand side of nodes represent dynamic parameter inputs to the domain-specific application while rectangles on the right hand side represent dynamic result data output. Data flows from source nodes to sink nodes as indicated by black arrows.

It is worth noting in Figure 1 that the interface is given in terms of the underlying science. Allowing the scientist to interact with the Grid in terms of their underlying science is vitally important to enabling the routine usage of Grid technologies.

We believe that visual programming environments that are customized for a particular scientific domain offer great promise in both increasing Grid usage and allowing the end scientists to focus on their science.

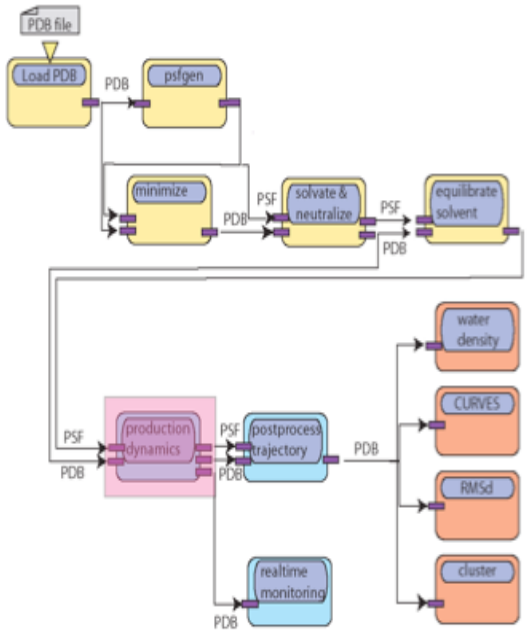

Figure 1